



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/900,060	07/06/2001	Travis J. Muhlestein	MSFT115921	7821
26389	7590	06/16/2004		
CHRISTENSEN, O'CONNOR, JOHNSON, KINDNESS, PLLC 1420 FIFTH AVENUE SUITE 2800 SEATTLE, WA 98101-2347			EXAMINER VU, TUAN A	
			ART UNIT 2124	PAPER NUMBER

DATE MAILED: 06/16/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/900,060

Applicant(s)

MUHLESTEIN ET AL.

Examiner

Tuan A Vu

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 06 July 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-17 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-17 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 7/6/2001 is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>09/24/2001</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the application filed July 6, 2001

Claims 1-17 have been submitted for examination.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1, 3-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Foody et al., USPN: 5,732,270 (hereinafter Foody) , in view of Katchabaw et al., "Making Distributed Applications Manageable Through Instrumentation", The Journal of Systems and Software 45, 1999 (hereinafter Katchabaw).

As per claim 1, Foody discloses a method for providing access to an external system software components from within a runtime computing system (Fig. 1), comprising:

providing an external system software components access interface within said runtime environment (e.g. *Proxy object, the System* – Fig. 1; Fig. 10; *proxies ... in process .. the system* - col. 19, lines 23-39);

receiving a request at said software components access interface for an external system software components available outside of said runtime environment (e.g. step 42 – Fig 8; *first object, how to access ... call its methods, set its properties, handle exceptions* - col. 8, lines 48-

Art Unit: 2124

61 – Note: methods, properties, exceptions read on an external system software components from outside the runtime environment);

transmitting a request for said an external system software components to an external system software components source external to said runtime environment (e.g. *intercepted by the system, the system forwards* - col. 8, lines 58-65; col. 17, lines 5-35; *COM server 78* – Fig. 12C; *OSA*, col. 9, lines 5-47 – Note: the system intercepting the proxy directions to access the real object via COM service is equivalent to transmitting a request for data to an external system software components external source);

receiving a response to said request (e.g. Fig. 9; *COM server 78* – Fig. 12C; *queries the objects, querying... information* -col. 10, lines 35-51; step 702 - Fig. 7C – Note: retrieving code or objects from additional service reads on receiving a response to request for object retrieval; and Expose class Factory and return results for Factory class construction reads on receiving response to request);

converting said response to a format compatible with said runtime environment (e.g. col. 13, line 8 to col. 14, line 49; Fig. 5,6-7b; step 704 – Fig. 7C); and

responding to said request with converted response (e.g. step 51, 52 – Fig. 8).

But Foody does not expressly disclose that the external system software components is instrumentation data. The providing of class objects and methods as tailored by Foody according to a proxy-derived description in order to provide objects components making up the required software project of a specific runtime application is similar to providing components of application instrumentation. In a system to provide class objects from an external source to a runtime by requesting developers via a OSI-based management agent similar to the OSA/System

Art Unit: 2124

by Foody, Katchabaw discloses providing instrumentation components via an agent for data access to the developers' application runtime environment (e.g. ch. 3-5, 6.1-2, pg. 82-93). It would have been obvious for one of ordinary skill in the art at the time the invention was made to modify to the multi-system code component assembling and customized delivery by Foody so to expand it into instrumentation components gathering and retrieval as taught by Katchabaw in order to support objects creation compliance, exceptions handling, and error checking as suggested by Foody (see Fig. 15) in order to provide a fault-free code for integration in the user's environment.

As per claim 3, Foody discloses converting object created in accordance with required configuration of runtime object (e.g. col. 15, line 8 to col. 16, line 28 – Note: factoring object according to specification of a dynamic proxy is equivalent to converting object model that is compatible with runtime environment of the requesting user; Fig. 6-7; col. 18, lines 7-36).

As per claim 4, Foody discloses returning a object (e.g. *Export Framework* – Fig. 2; col. 7, lines 39-59), a Factory (col. 16, lines 48-64; col. 17, line 55-66 – Note: creating a object by OSA from putting together classes and methods as in Fig. 5-7, 13-14, is equivalent to packaging properties of management object through instrumentation data source)

As per claims 5 and 6, Foody discloses exception object (e.g. *exceptions* - col. 10, lines 10-25; *VExceptionData* – Fig. 15; step 707 – Fig. 7C; *kVUsageProtected*, *kVUsageDoNotCache* – Fig. 15).

As per claims 7 and 8, Foody discloses a computer-readable medium (col. 235, lines 61-64)

As per claim 9, Foody discloses a method for accessing an external system software components from within a runtime computing environment (Fig. 1), comprising:

receiving a request to construct a management object comprising said system software components(step 42 – Fig 8);

in response to said request, querying an instrumentation data access interface within said runtime environment for said software components (e.g. Fig. 2; com Dll ‘in Process’ – Fig. 12a);

determining whether said software components was successfully returned (Fig. 11-12 – Note: remote calls to COM server inherently discloses determining whether software components being retrieved are successfully returned; step 702-703 Fig 7C; *return errors* - col. 12, lines 39-47); and

in response to said successful return, constructing said management object and populating said object with requested software components (e.g. Fig. 7C, 9, 13-15 – Note: traversing class to create object class and methods for creating a object from the Factory is equivalent to constructing an management object and populating it with software objects).

But Foody does not expressly disclose that the software components requested from the runtime environment are instrumentation data; but this limitation has been addressed in claim 1 above.

As per claim 10, Foody discloses binding a management class to an instance of management object (Fig. 8, 13-15)

As per claim 11, Foody discloses identifying a Namespace (e.g. col. 10, lines 10-39)

As per claim 12, Foody discloses management path object (e.g. *path ...information 68* – Fig. 10; *location ...framework* – Fig. 2)

As per claim 13, Foody discloses interface calls options to access class objects (e.g. *DSOM*, *SOM*, *COM* - col. 9, lines 51-61; CORBA, DII – col. 10, lines 32-48 – Note: providing different ways to access object is equivalent to providing access connecting options)

As per claim 14, Foody discloses exception when constructing a class; but does not provide throwing a management exception; but in view of the creating of class to access data using object-oriented programming constructs (see Appendix VViewNameSpace, Exception - col. 61-126), the suggestion as to throw exception upon non-success using class definition to access a management object, e.g. the namespace, is shown. Official notice is taken that using object-oriented like Java to implement access of data via a communication link and thus throw an exception call when a communication occurs was a well-known concept at the time the invention was made. Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide code as suggested by Foody so to implement communication with the management service by COM or OSA such that exception are thrown when such communicating incurs errors as suggested in Foody's Appendix and taught by well known practices.

As per claim 15, see Foody Appendix (e.g. *VcrCallException*, *Index* - col. 53-54)

As per claims 16 and 17, refer to the rationale of claims 7-8, respectively.

4. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Foody et al., USPN: 5,732,270, and Katchabaw et al., " Making Distributed Applications Manageable Through Instrumentation", as applied to claim 1, and further in view of Festor et al., "Integration of WBEM-based Management Agents in the OSI Framework", Integrated Network

Management, 1999, Proceedings of the 6th IFIP/IEEE International Symposium (hereafter Fester).

As per claim 2, Foody discloses a runtime API for data access (col. 11, lines 15-25) and remote call for COM access (Fig. 12C) and Dynamic Invocation interface (col. 12, lines 48-59); but does not explicitly shows a runtime client API. The use of windows-based system APIs to effect communicating invocations between layers of application codes and to make outgoing calls to services to retrieve data was a known concept in the art at the time of the invention. Fester, in a system using a management service to provide objects to a developing environment analogous to Foody's method, discloses client Java communication interfaces (e.g. Fig 3; ch. 7.1, pg. 59-61). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide the user's runtime environment with Java application interface as taught by Fester, in case the runtime environment by Foody is specific to an external Web-application that is not a heterogeneous application, so that by using the Java client APIs at the web-based client site as taught by Fester, the resources for creating a instant proxy by Foody would be alleviated and the process of retrieving code components for the specific Web-based application instrumentation would be more efficient because Java APIs are widely known to be readily available in many platforms compatible and operable with Web-based applications at the time the invention was made.

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

Art Unit: 2124

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please label

"PROPOSED" or "DRAFT" – please consult Examiner before use)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive,
Arlington. VA. , 22202. 4th Floor (Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding
should be directed to the receptionist whose telephone number is (703) 305-3900.

Kakali Chaki

VAT
June 6, 2004

**KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100**